

# 实验11实验报告

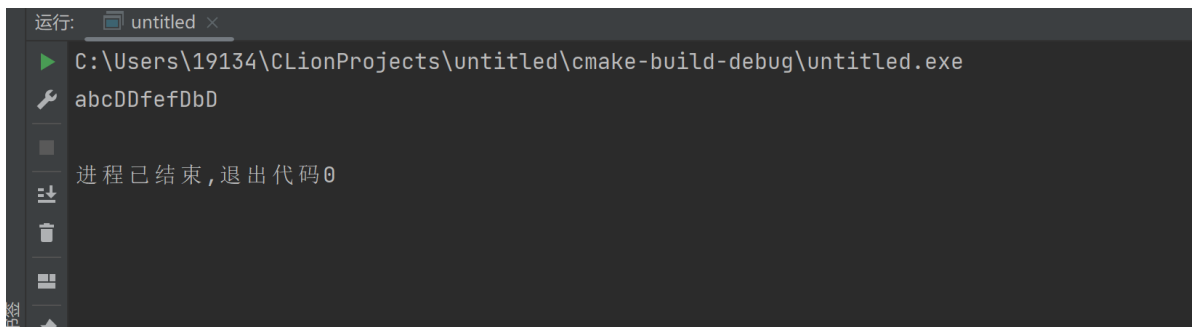
---

## 实验内容1

### 习题三 (3)

经调试，该程序中ss函数的作用是将字符数组s中的字符t全部由小写变为大写

运行结果如下：

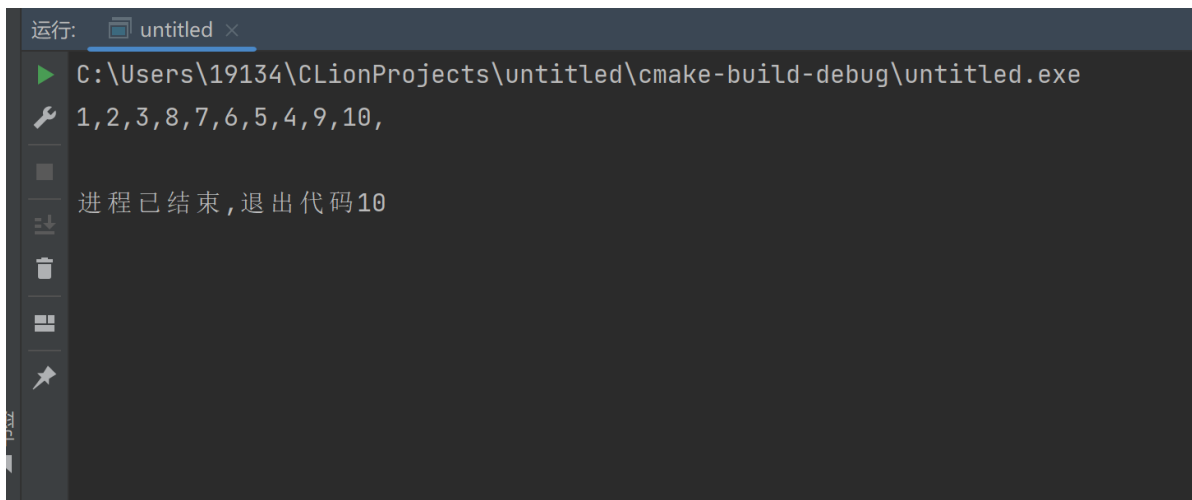


```
运行: untitled x
C:\Users\19134\CLionProjects\untitled\cmake-build-debug\untitled.exe
abcDDfefDbD
进程已结束,退出代码0
```

### 习题三 (6)

sort函数的作用是，从传入的数组开始，将一共n个数字进行从大到小排序

运行结果如下：



```
运行: untitled x
C:\Users\19134\CLionProjects\untitled\cmake-build-debug\untitled.exe
1,2,3,8,7,6,5,4,9,10,
进程已结束,退出代码10
```

---

## 实验内容2

### 实验步骤 (1)

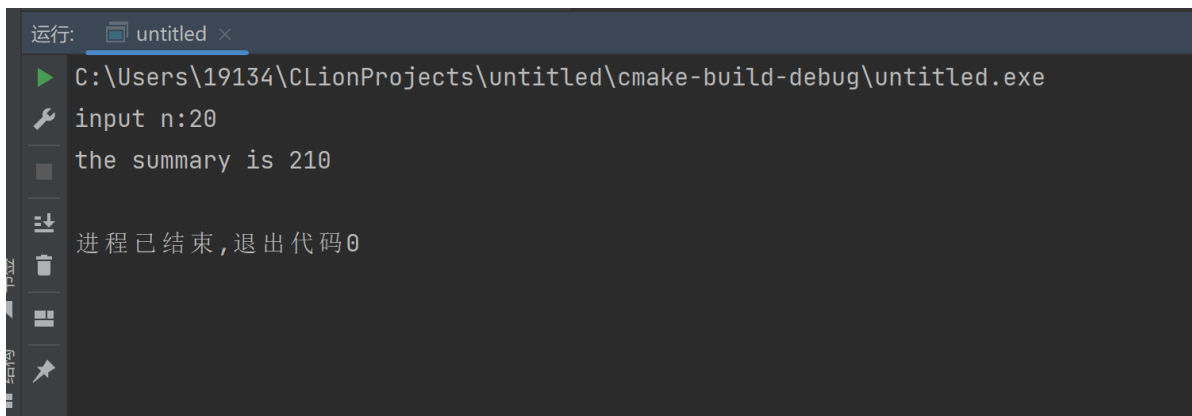
代码编写如下：

```

#include <stdio.h>
int psum(int n) {
    int i, sum = 0;
    for (i = 0; i <= n; i++)
        sum += i;
    return sum;
}
int main() {
    int n;
    printf("input n:");
    scanf("%d", &n);
    printf("the summary is %d\n", psum(n));
}

```

运行结果如下图：



```

运行: untitled x
C:\Users\19134\CLionProjects\untitled\cmake-build-debug\untitled.exe
input n:20
the summary is 210
进程已结束,退出代码0

```

## 实验步骤 (2)

代码编写如下：

```

#include <stdio.h>
void max_min(double *num, double *max, double *min) {
    int n;
    for (n = 1; n < 10; n++) {
        if (num[n] > *max)
            *max = num[n];
        if (num[n] < *min)
            *min = num[n];
    }
}
int main() {
    double num[10], max, min;
    int i;
    for (i = 0; i < 10; i++)
        scanf("%lf", &num[i]);
    max = num[0], min = num[0];
    max_min(num, &max, &min);
    printf("the maximum number is %lf\nthe minimum number is %lf\n", max, min);
}

```

运行结果如下图：

```
运行: untitled x
C:\Users\19134\CLionProjects\untitled\cmake-build-debug\untitled.exe
889.67
1.23
0.001
2.67
19.28
3.1415926
123
666.7
1.22
0.1
the maximum number is 889.670000
the minimum number is 0.001000

进程已结束,退出代码0
```

### 实验步骤 (3)

代码编写如下:

```
#include <stdio.h>
#include <math.h>
short regress(int num) {
    int digit, sum = 0, num1 = num;
    for (digit = 0; num1 != 0; digit++) //计算出num的位数, 存储在digit中
        num1 /= 10;
    num1 = num;
    while (num1 != 0) {
        sum += pow(num1 % 10, digit);
        num1 /= 10;
    } //计算sum, 随后判断是否满足定义
    if (sum == num)
        return 1;
    else
        return 0;
}
int main() {
    int num;
    for (num = 100; num < 1000; num++)
        if (regress(num))
            printf("%d\n", num);
    return 0;
}
```

运行结果如下:

